

# Take your IBM i development to the next level with MERLIN

*By Randy Ruhlow*

In today's ever-evolving world, businesses are challenged with keeping up with constantly evolving technologies, particularly due to the sheer volume of these interconnected electronic devices and computing platforms. Businesses have to meet the increased demands, both internally and those of clients, that the internet of things places on their business applications and data. But it is not only the burden of these demands, it is also the rate at which these changes are required for a business to remain functionally optimal and competitive.

Today's IT and software development teams are evolving their development, delivery and deployment approaches to be more responsive, reactive and efficient. Essentially, they are evolving into DevOps, which is an agile, automated set of software management processes, cultural behaviors, tooling and practices. DevOps elevates the frequency of software delivery, creates higher-quality software, reduces risks and enhances overall responsiveness to ever-changing business demands. IBM i can participate in and benefit from this accelerated DevOps approach to its software lifecycle management systems. Welcome to IBM i's Modernization Engine for Lifecycle Integration – MERLIN, IBM i's newest product to assist clients with IBM i application modernization and DevOps pipeline process development.

## What is MERLIN?

MERLIN is a browser-based application development and modernization platform for IBM i users, and this product is the first of its kind for IBM i. It provides a set of tools that run in Red Hat OpenShift containers and provides a framework to guide developers in the modernization of IBM i applications. It is based on the interoperability between IBM i and the Red Hat OpenShift container-based functions.

At the initial release of MERLIN, the platform introduced IBM i Developer and IBM i CI/CD, two tools designed to provide a modern software development and devops CI/CD (continuous integration/continuous deployment) pipeline environments. Both tools utilize Git and the IBM i CI/CD integrates with an internal Jenkins automation server.

ARCAD, an IBM partner, was integral to the development of the container-based model in which these application modernization tools are housed. ARCAD has been in business for more than 30 years and has provided the IBM i community with application lifecycle management and modernization tools across business and technology sectors.



# What does MERLIN do for the IBM i platform?

MERLIN helps improve the IBM i platform by enabling developers to easily interact with the applications, supporting developers with state-of-the-art interfaces for performing modernization and updating of the applications. Security is always of the utmost importance; the tools ensure that applications are written securely across the entire development lifecycle through built-in automated security and testing features.

MERLIN supports both new and older versions for RPG code. Newer developers may find dealing with fixed-formatted RPG challenging as most other coding languages are in a free-form format. This tool solves the problem by automatically converting fixed-format RPG code to free-form RPG code so any developer can comfortably work with this code. This feature makes the IBM i platform more accessible to different types of developers with varied experiences.

MERLIN leverages two open source tools, which may appeal to newer developers. Git, the source control management (SCM) system and Jenkins, an automation server for DevOps pipelines are both highly leveraged into the MERLIN toolsets.

The IBM i Developer integrated development environment (IDE) is browser-based and provides easy access to a robust IBM i coding development environment. The code-ready IDE allows for consistency, quick deployment and ease of access to IBM i development environments

The IBM i CI/CD provides a GUI driven DevOps Jenkins pipeline generator and has the ability to drive multi-stage tasks in one pipeline. A typical pipeline consists of SCM, build, test and deploy stages.

Developers will like the IBM i application development CI/CD feature. The portability feature that Red Hat OpenShift containers provide will enable developers to easily move to different environments and clouds as well.

# Core features of the MERLIN platform and initial tools

The MERLIN platform consists of several core features to support centralized management for MERLIN and the tools offered through its catalog.

**1. The basic functions:** Several core features comprise the base functions that support the centralized management for MERLIN. Regarding the overall management of MERLIN, a monitoring capacity monitors the total resources of MERLIN. In line with this, a capacity for tool lifecycle management is responsible for deploying, upgrading and deleting MERLIN tools.

Users are managed through a tool that creates, modifies and deletes MERLIN users and their authorities. MERLIN users are authenticated through a single sign-on capability.

The overall security of the platform's architecture is addressed in a few ways.

- Inventory management is performed by the storage and management of connecting systems information with which MERLIN interacts
- MERLIN securely stores and manages the credentials for the interconnected systems
- MERLIN provides and manages certificates for TLS-based connections among containers and other systems
- Certification management is done by creating and managing certifications for TLS-based connections among containers and other systems

Regarding other machines or infrastructure, IBM i virtual machines are provisioned, managed or deleted through the IBM i VM Management feature. Any necessary software packages that MERLIN requires can be installed via the IBM i Software Installer. Finally, RESTful APIs can be created or managed against IBM i programs and data stored in Db2 for i.

## **2. IBM i Developer—Integrated Developer Environment (IDE):**

This is a development environment that provides an in-browser IDE developers can use to develop IBM i applications from any machine. This provides a single-click code-ready developer workspace and eliminates local environment configurations. The IDE provides enterprise readiness with built-in security.

The developer benefits from having access to a modern editing experience for IBM i languages, which include RPG, SQL and others. A developer can perform the following actions: tokenization; formatting, content assist, code folding, hover information, outline view, refactoring, error reporting and more.

Developers can easily interact with IBM i objects and the Integrated File System (IFS) files, while having support for project-based developer builds. Developers can access modern source control due to the integration of Git and the ARCAD tools.

## **3. IBM i CI/CD - Continuous Integration/Continuous Deployment:**

IBM i CI/CD aims to simplify the experience of DevOps for IBM i application development. Developers can leverage out-of-the-box Jenkins with the ARCAD integration. With this, users can create their own Jenkins environment with capability to build and deploy IBM i programs. MERLIN provides a graphic interface for the developer, which simplifies the key operations of Jenkins. This interface creates Jenkins pipelines specifically for IBM i programs development.

# MERLIN's architecture: An inside look

The first view of MERLIN shown in Figure 1 displays the internal component composition of MERLIN and the external interfaces and connections into MERLIN. A typical MERLIN implementation will include one or more workstations, a Git source, and one or more systems – and in this case they happen to be IBM i systems. Each of these three components are externally connected to the MERLIN platform and its tools.

MERLIN relies on a network-oriented architecture as Figure 1 demonstrates. There are many good reasons for moving to a containerized environment such as a Red Hat OpenShift Container Platform (RH OCP); and for MERLIN, it takes advantage of the cluster's application deployment capabilities to handle the complexities of establishing all of the internal network connectivity required each time a new IDE or CI/CD portal is instantiated. This leaves a more normal network exercise for connecting the external components to MERLIN.

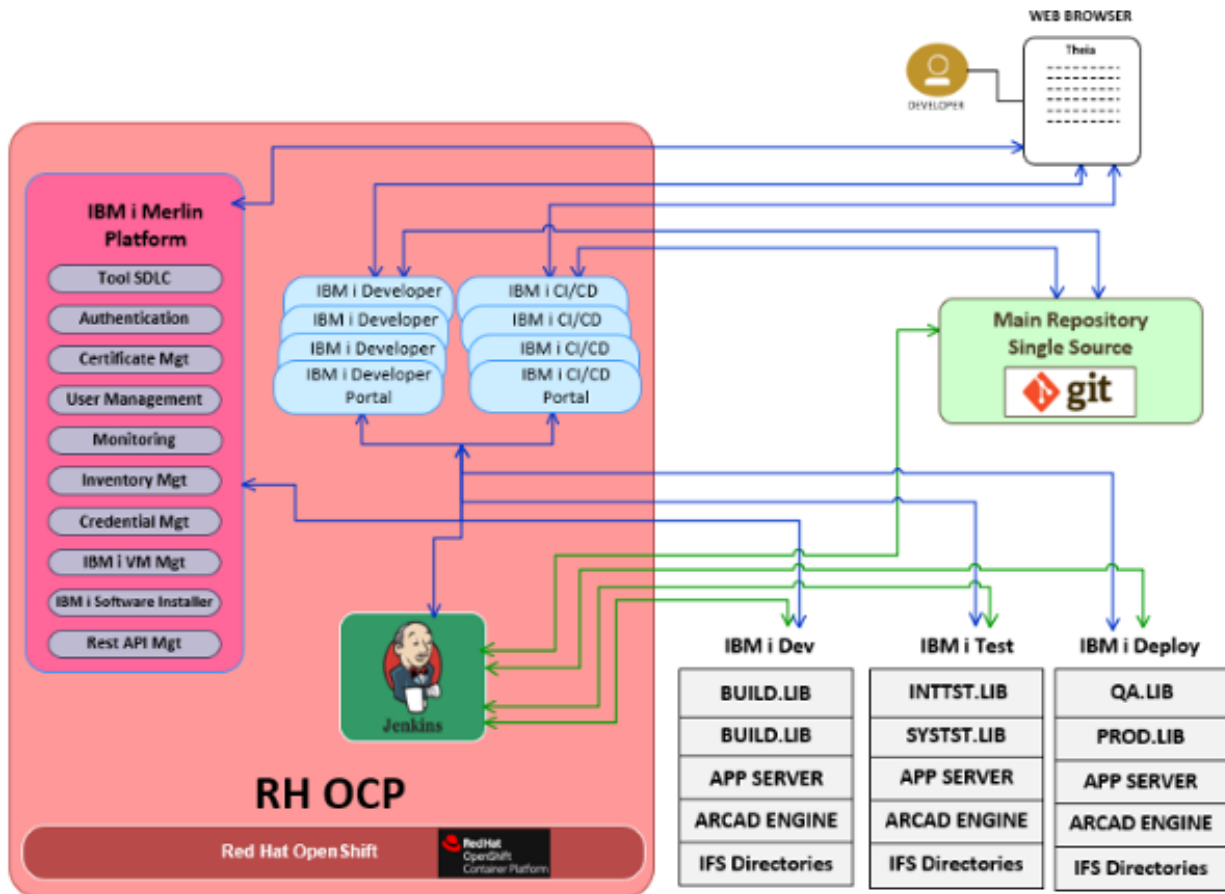


Figure 1. MERLIN at a high level

The next sections explore different aspects for insight into the overall architecture of MERLIN.

## MERLIN's containerized components

MERLIN is an IBM certified container-based platform offering multiple IBM i offerings. Figure 2 displays the MERLIN platform and its two initial IBM i tool offerings all existing in a RH OCP cluster. Included in the MERLIN platform's IBM i CI/CD tool is an internal containerized Jenkins server. These four components comprise the core MERLIN offering.

Keep in mind that each of these components are running in one or more containers on worker nodes for the RH OCP cluster and will scale based on demand. This is another benefit of containerized applications.

1. The IBM i MERLIN Platform is the base that provides functions to support centralized management for MERLIN and its IBM i tools. The core capabilities are listed in Figure 2.
2. The IBM i Developer is the web-based IDE for developers to develop, analyze and build IBM i programs and applications. It is integrated with ARCAD plugins.
3. The IBM i CI/CD provides an IBM i focused GUI pipeline interface connected with a containerized Jenkins server, and it is integrated with ARCAD plugins.
4. An integrated Jenkins server provides the underlying pipeline capabilities.

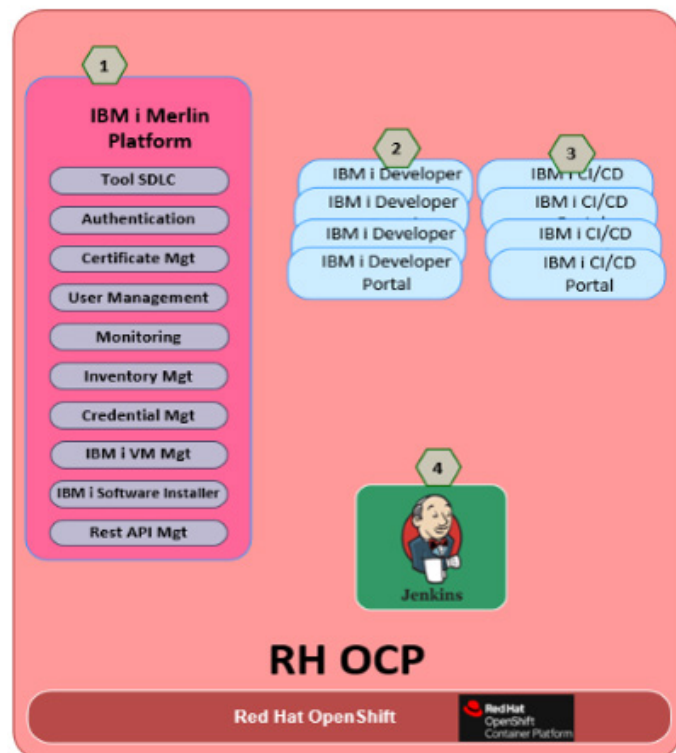


Figure 2. MERLIN Components

## MERLIN's external interfaces

All access to MERLIN and its tools is done via web browser interfaces. Installing the MERLIN platform along with the IBM i Developer and IBM i CI/CD tools will create three base URL access points. These access points allow for the configuration of the MERLIN platform, the installation and configuration of the IBM i Developer and IBM i CI/CD portals, and the launching of instances of these tools and their workspaces. These links persist through container recycling events and can be bookmarked for ease of access.

Figure 3 displays the three base interface accesses plus a fourth link for access to the internal Jenkins server. The Jenkins server is created after you install and configure the IBM i CI/CD tool. The CI/CD tool provides a wizard to create “tasks,” which are an abstraction layer above the pipelines that are generated on the Jenkins server. Tasks provides the ability for developers to easily generate “pipelines” without having to have intimate knowledge of Jenkins and pipeline scripting.

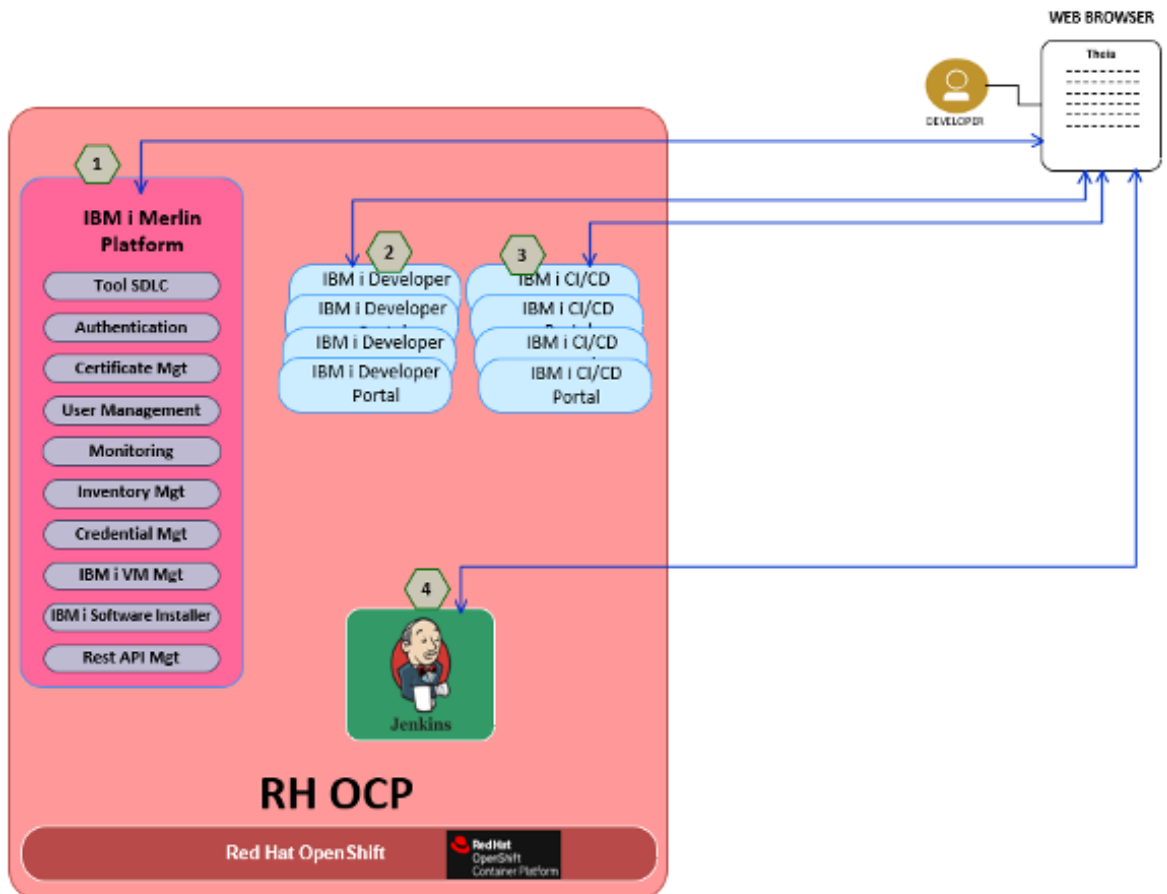


Figure 3. MERLIN Interfaces



## External IBM i connections

MERLIN provides the IBM i systems with VSCode-based IDE application development workspaces, and CI/CD pipelines for automating the building, testing and deploying of their applications. This integration requires a Git repository used for source control of the application artifacts and code, and an automation server conveniently provided by an internal, containerized Jenkins server run on the RH OCP cluster. While the RH OCP provides for the interconnectivity between the components living in the cluster, there are a handful of connections required for a full MERLIN DevOps environment. Figure 4 identifies those external connections for the IBM i systems.

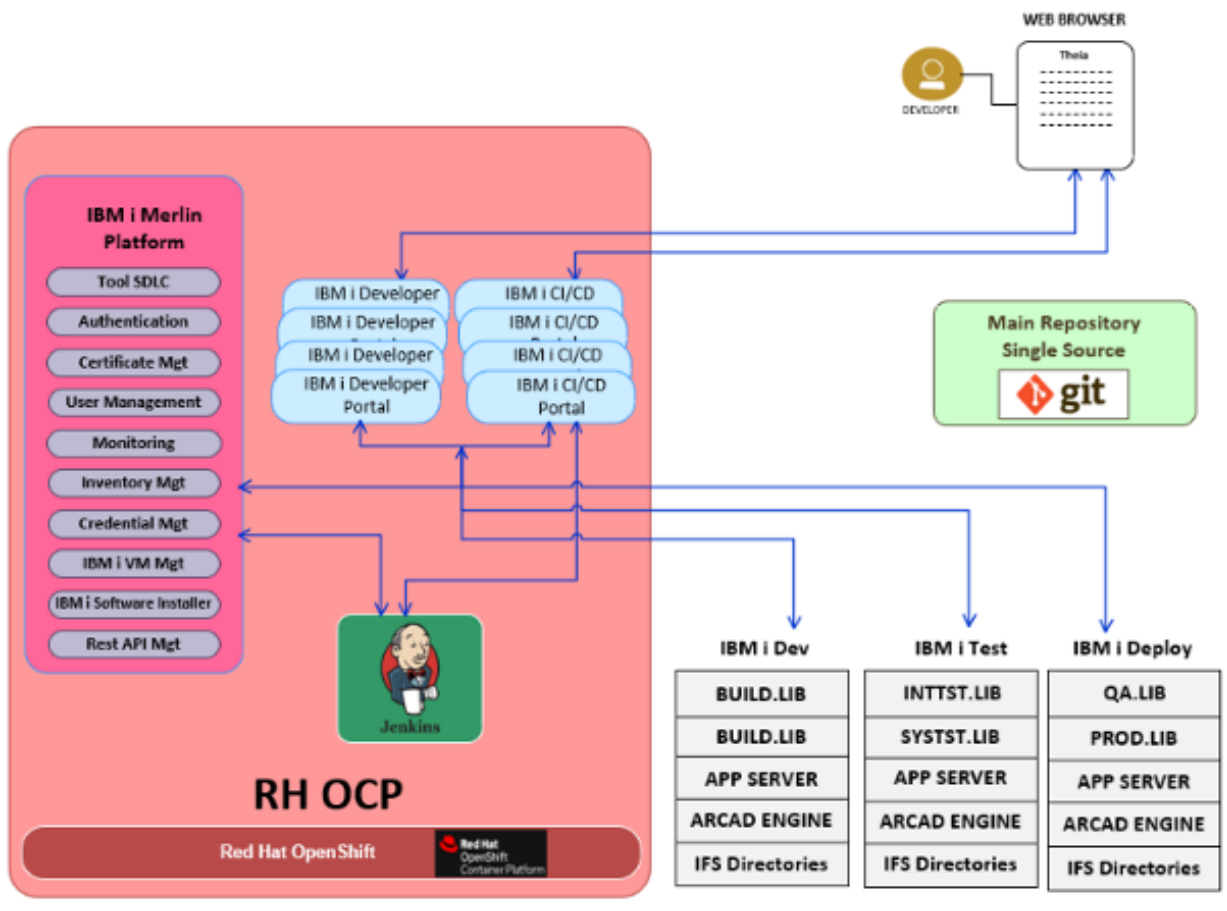


Figure 4. MERLIN IBM i Connections

## Git connections

The Git server is the last of the external connections required for the MERLIN DevOps implementations. It provides the single-source code repository or SCM for your applications. It is a tried and true source control management tool and is integrated with a majority of the state-of-the-art DevOps solutions. It works well with the Jenkins automation server having a feature of “Git-hooks” used for sending notifications when actions occur on a given Git repository.

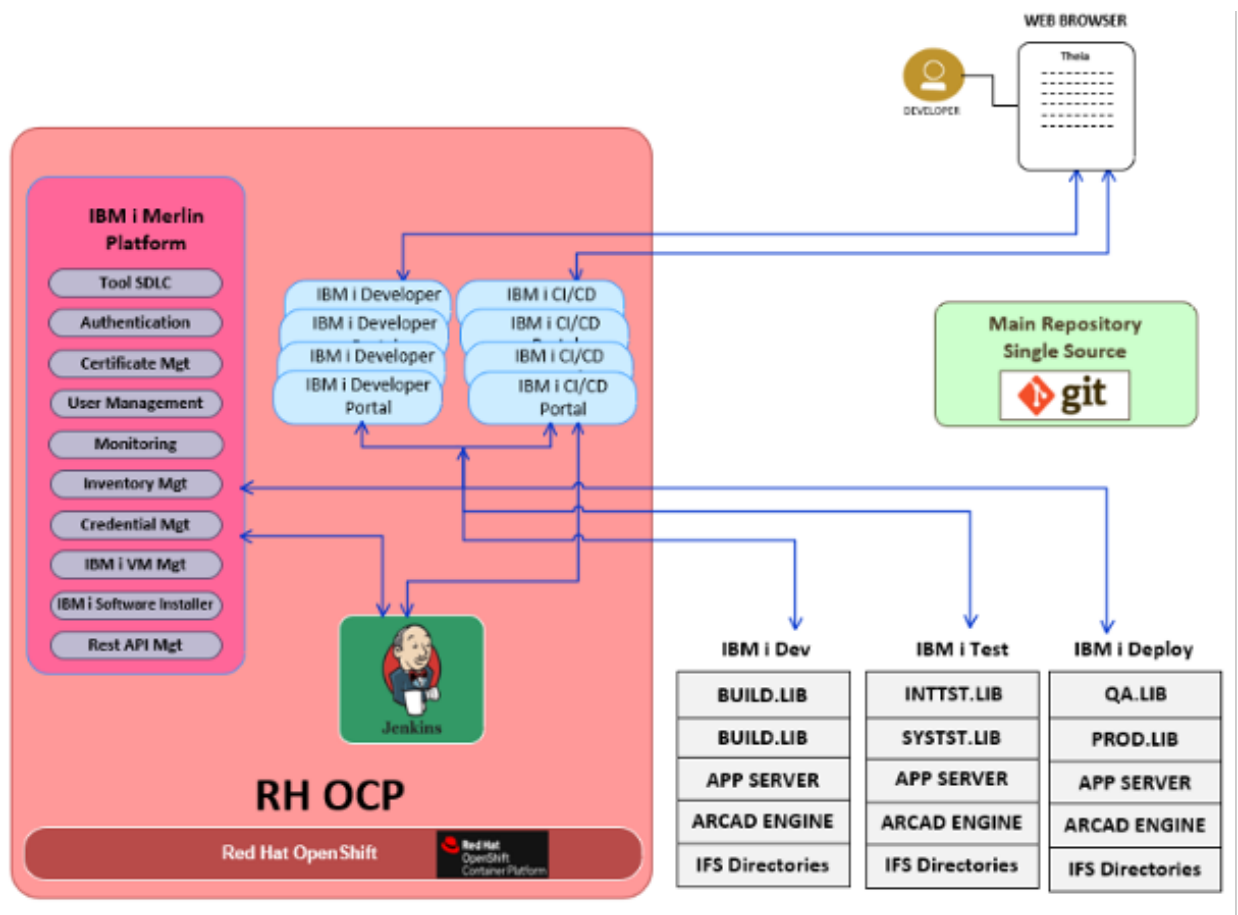


Figure 5. MERLIN Git Connections

We have finished our brief walk-through of the MERLIN architecture. We will move on to what MERLIN provides you with in regards to simplifying your IBM i build cycles.

# The advantages of using MERLIN versus a traditionally tooled CI/CD

So how does MERLIN fit within an IBM i DevOps environment? Let's first review a common, basic IBM i traditional continuous integration (CI) reference diagram laid out in Figure 6. In this scenario, we are using A-GitHub enterprise repository server; B-Rational Developer for I IDE on the developer workstation; C-locally cloned branch of the application repository; D-an IBM i development server; and E-Jenkins automation server for processing GitHub-triggered events.

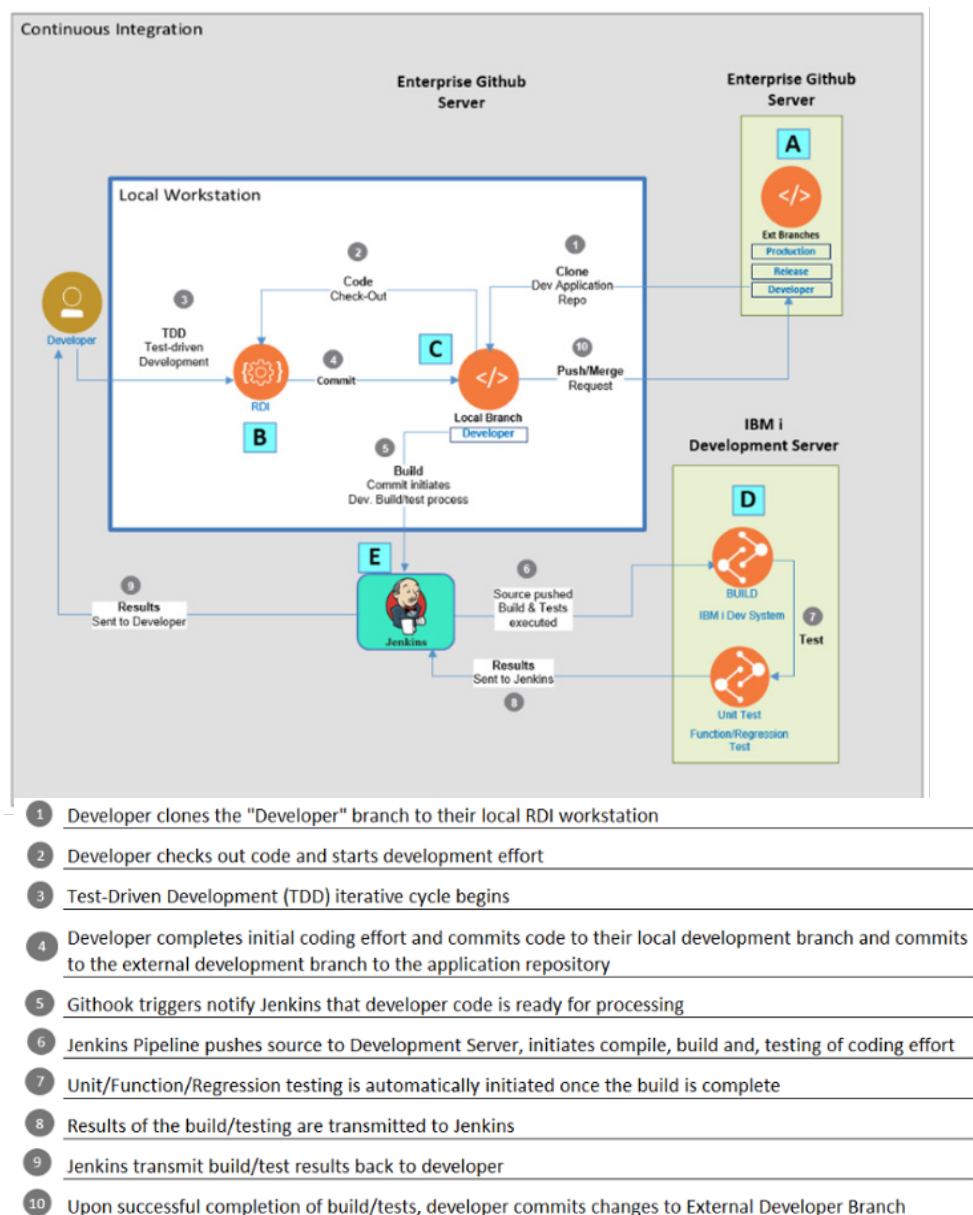


Figure 6. IBM i Continuous Integration

The normal development cycle begins when the developer clones a development branch of an external application repository to their workstation and commences making the requisite code changes dictated by the project. Once completed, the developer commits the code back into their development branch of the repository. This covers items 1-4 in Figure 6.

It is at this point that the automation kicks in. GitHub broadcasts a notification when the code is checked into the development branch on the external application repository, and because the Jenkins server is configured to listen for this broadcast, causes the Jenkins server to launch a pipeline to process the changes. This covers the steps of the process in items 5-9 in Figure 6.

This sounds simple, but there are required scripts and configurations that were manually created to make this all work, both for the Jenkins and IBM i servers. The following is an abbreviated list of the scripting tasks:

1. Write the Jenkins script that performs the following:
  - a. Copy the changed code in the development branch from the external application repository to the IBM i development library
  - b. Determine all compile dependencies are available for the compilation of the changed code
  - c. Execute the compile and build of the changed code
  - d. Execute the testing of the changed code
  - e. Report the results of the compile and test actions
2. Write the IBM i dependency, build and test scripts that will be executed in step 1.

Let's compare and contrast that with a MERLIN CI/CD implementation. First, MERLIN uses the same tools as the traditional approach. In fact, the infrastructure architecture for MERLIN is very similar to the one in Figure 6. There is no need to redraw it for MERLIN at this time. The source code is stored in GitHub and the compiles, builds and testing occur on an IBM i development system. The IBM i compile, build and testing commands and/or scripts will be the same that are called in the traditional pipeline, and the pipeline used in MERLIN will run on an internal Jenkins server.

The main difference between the two approaches is how the pipelines were developed. This is the key benefit of MERLIN CI/CD. Unlike the traditional approach, MERLIN's pipeline was created using a GUI wizard to develop the pipeline and the context of its different stages. This allows the developer to become more efficient and gives them the ability to create workable pipelines to automate their integration without having to have intimate knowledge about the Jenkins server or its scripting languages.

Some examples of the wizard's screens are shown in Figure 7:

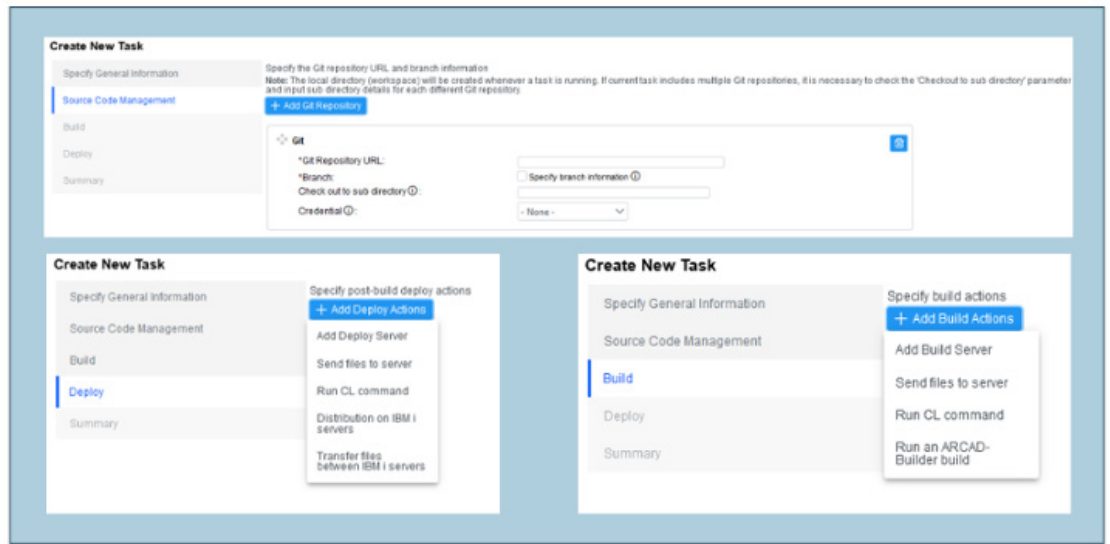


Figure 7. MERLIN Wizard CI/CD Tasks Screen Shots

The “Create New Task” wizard provides options to connect and clone Git repositories, connect to build servers, and execute specific functions on the IBM i server for the compile and build of the code or ARCAD build server, and connect to deploy servers with options that facilitate the deployment of the newly built code.

The output of the generated pipeline is produced on the internal Jenkins server and is executed when initiated from MERLIN CI/CD. MERLIN allows for the configuration and use of an external Jenkins server if you desire to make use of existing build and deploy artifacts.

Figure 8 depicts the flow of a MERLIN generated high-level pipeline demonstrating some of its capabilities.

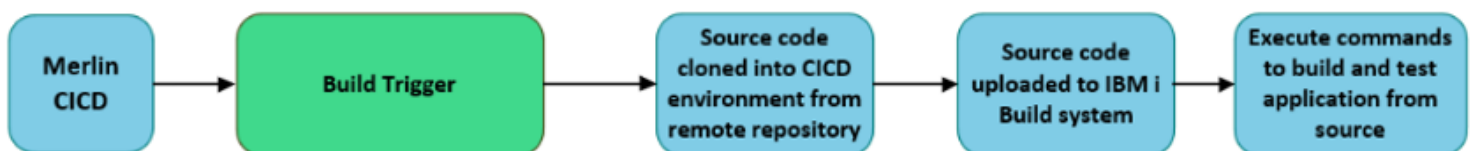


Figure 8. MERLIN-Generated CI/CD Pipeline



## Learn more

This wraps up our closer look at IBM i's MERLIN and CI/CD, and how it can help streamline and accelerate your modernization efforts. To learn more about how you can make MERLIN work for you or IBM Technology Services, visit:

[ibm.com/services/infrastructure](https://ibm.com/services/infrastructure)

or email us at:

[technologyservices@ibm.com](mailto:technologyservices@ibm.com)

*Randy Ruhlow is a senior managing consultant for IBM Technology Services specializing in IBM i DevOps and Complex Systems Services.*

---

© Copyright IBM Corporation 2023

IBM Corporation  
Route 100  
Somers, NY 10589

Produced in the United States of America  
January 2023

IBM, the IBM logo, ibm.com, GDPS, IBM zSystems, LinuxONE and Power are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Red Hat® and OpenShift® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.



Please Recycle

## Additional references

Getting started for users YouTube videos: <https://www.youtube.com/playlist?list=PLPELYviDwCnY6L5r5ZnmCneqhakLcB7ko>

Getting started for administrators YouTube videos: <https://www.youtube.com/watch?v=6UIN1vhrCPo&list=PLPELYviDwCnajC310KB2K7zfu51FPuYJS>

MERLIN documentation on Github: <https://ibm.github.io/merlin-docs/#/>

IBM MERLIN documentation: <https://www.ibm.com/docs/en/merlin/1.0>

Test Drive MERLIN: <https://ibm.github.io/merlin-docs/#/./guides/overview/sandbox>